

5 基础控件

万永权

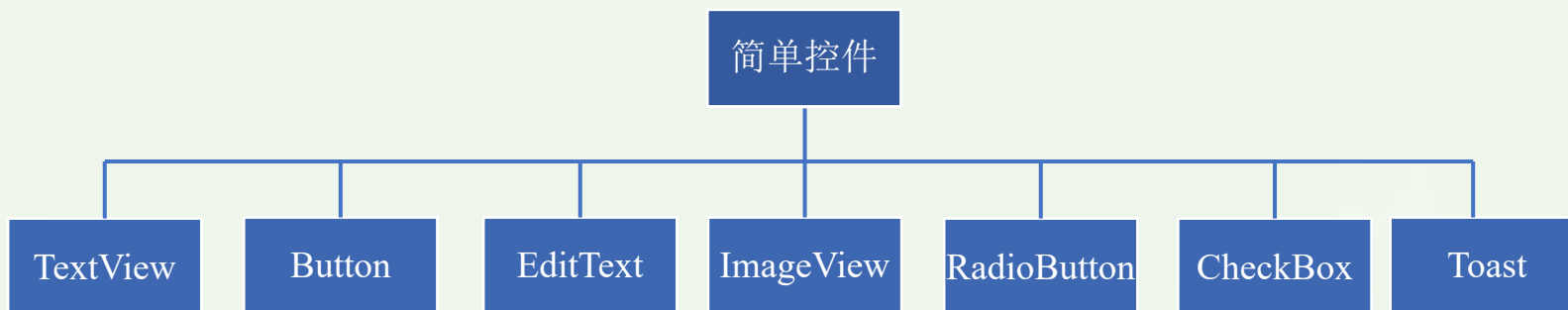
// 学习目标/Target



□ 掌握简单控件的使用，能够独立搭建一个注册界面

5.2 界面控件

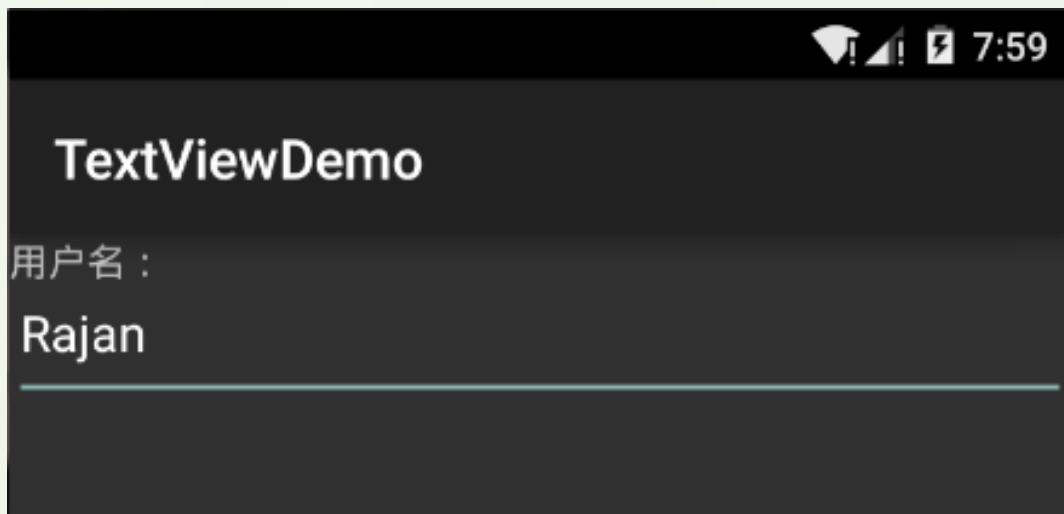
控件是界面组成的主要元素，为了显示界面上的输入框、图片、文字等信息，Android系统提供了一些控件来显示这些信息，每个控件都有对应的属性来设置不同的效果。我们以控件使用的复杂程度将Android中的控件分别简单控件和列表控件，简单控件包含以下几种，具体如下图所示。



5.2 界面控件

• 5.2.1 TextView和EditText

- TextView是一种用于显示字符串的控件
- EditText则是用来输入和编辑字符串的控件
- EditText是一个具有编辑功能的TextView



// TextView控件



TextView控件用于显示文本信息，我们可以在XML布局文件中以添加属性的方式来控制TextView控件的样式，[TextView控件的属性](#)如下表所示。

TextView控件的属性

属性名称	功能描述
android:layout_width	设置TextView控件的宽度
android:layout_height	设置TextView控件的高度
android:id	设置TextView控件的唯一标识
android:background	设置TextView控件的背景
android:layout_margin	设置当前控件与屏幕边界或周围控件、布局的距离
android:padding	设置TextView控件与该控件中内容的距离
android:text	设置文本内容
android:textColor	设置文字显示的颜色
android:textSize	设置文字大小，推荐单位为sp

// TextView控件



TextView控件的属性

属性名称	功能描述
android:gravity	设置文本内容的位置
android:maxLength	设置文本最大长度，超出此长度的文本不显示
android:lines	设置文本的行数，超出此行数的文本不显示
android:maxLines	设置文本的最大行数，超出此行数的文本不显示。
android:ellipsize	设置当文本超出TextView规定的范围的显示方式。
android:drawableTop	在文本的顶部显示图像
android:lineSpacingExtra	设置文本的行间距
android:textStyle	设置文本样式，如bold（粗体），italic（斜体），normal（正常）

// EditText控件



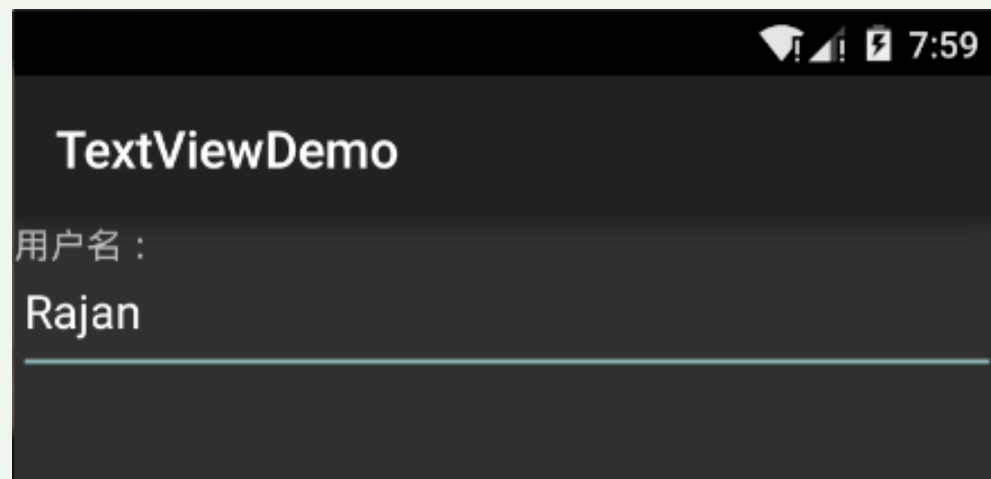
EditText表示编辑框，它是TextView的子类，用户可在此控件中输入信息。除了支持TextView控件的属性外，EditText还支持一些其它的常用属性，这些常用属性如下表所示。

属性名称	功能描述
android:hint	控件中内容为空时显示的提示文本信息
android:textColorHint	控件中内容为空时显示的提示文本信息的颜色
android:password	输入文本框中的内容显示为 “.”
android:phoneNumber	设置输入文本框中的内容只能是数字
android:maxLines	设置文本的最大行数
android:scrollHorizontally	设置文本信息超出EditText的宽度情况下，是否出现横拉条
android:editable	设置是否可编辑

5.2 界面控件

• 5.2.1 TextView和EditText

- 建立一个“TextViewDemo”的程序，包含TextView和EditText两个控件
- 上方“用户名”部分使用的是TextView，下方的文字输入框使用的是EditText



5.2 界面控件

• 5.2.1 TextView和EditText

- TextViewDemo在XML文件中的代码

```
<TextView android:id="@+id/TextView01"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="TextView01" >  
</TextView>  
<EditText android:id="@+id/EditText01"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="EditText01" >  
</EditText>
```

5.2 界面控件

• 5.2.1 TextView和EditText

- 第1行android:id属性声明了TextView的ID，这个ID主要用于在代码中引用这个TextView对象
 - “@+id/TextView01”表示所设置的ID值
 - @表示后面的字符串是ID资源
 - 加号（+）表示需要建立新资源名称，并添加到R.java文件中
 - 斜杠后面的字符串（TextView01）表示新资源的名称
 - 如果资源不是新添加的，或属于Android框架的ID资源，则不需要使用加号（+），但必须添加Android包的命名空间，例如
android:id="@android:id/empty"

5.2 界面控件

• 5.2.1 TextView和EditText

- 第2行的android:layout_width属性用来设置TextView的宽度，wrap_content表示TextView的宽度只要能够包含所显示的字符串即可
- 第3行的android:layout_height属性用来设置TextView的高度
- 第4行表示TextView所显示的字符串，在后面将通过代码更改TextView的显示内容
- 第7行中“match_content”表示EditText的宽度将等于父控件的宽度

5.2 界面控件

• 5.2.1 TextView和EditText

- TextViewDemo.java文件中代码的修改

```
TextView textView = (TextView)findViewById(R.id.TextView01);  
EditText editText = (EditText)findViewById(R.id.EditText01);  
textView.setText("用户名 :");  
editText.setText("Rajan");
```

- 第1行代码的findViewById()函数能够通过ID引用界面上的任何控件，只要该控件在XML文件中定义过ID即可
- 第3行代码的setText()函数用来设置TextView所显示的内容

5.2 界面控件

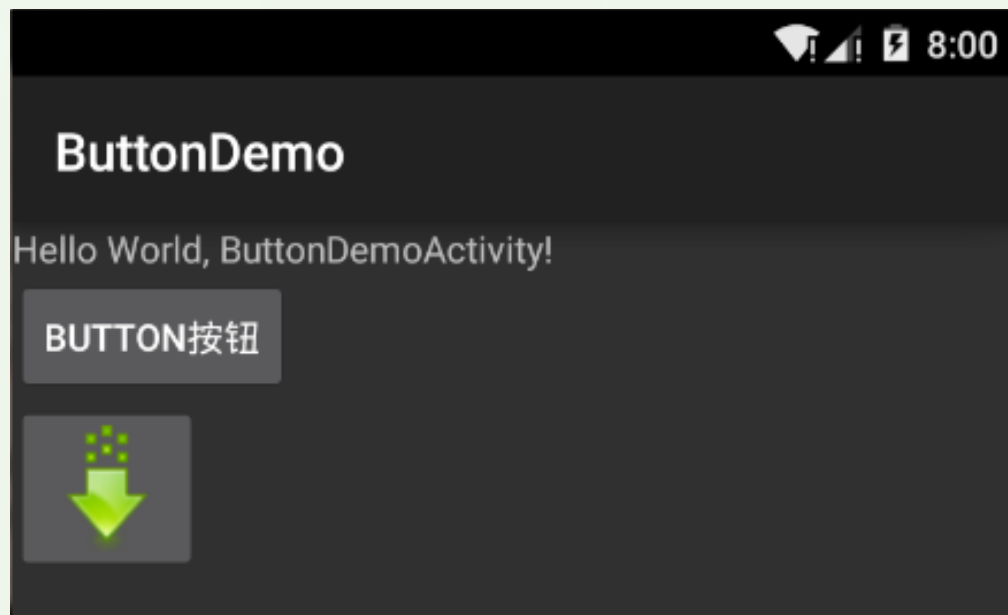
• 5.2.2 Button和ImageButton

- Button是一种按钮控件，用户能够在该控件上点击，并后引发相应的事件处理函数
- ImageButton用以实现能够显示图像功能的控件按钮

5.2 界面控件

• 5.2.2 Button和ImageButton

- 建立一个“ButtonDemo”的程序，包含Button和ImageButton两个按钮，上方是“Button按钮”，下方是一个ImageButton控件



5.2 界面控件

• 5.2.2 Button和ImageButton

- ButtonDemo在XML文件中的代码

```
1. <Button android:id="@+id/Button01"  
2.   android:layout_width="wrap_content"  
3.   android:layout_height="wrap_content"  
4.   android:text="Button01" >  
5. </Button>  
6. <ImageButton android:id="@+id/ImageButton01"  
7.   android:layout_width="wrap_content"  
8.   android:layout_height="wrap_content">  
9. </ImageButton>
```

- 定义Button控件的高度、宽度和内容
- 定义ImageButton控件的高度和宽度，但是没定义显示的图像，在后面的代码中进行定义

5.2 界面控件



• 5.2.2 Button和ImageButton

- 引入资源
 - 将download.png文件拷贝到/res/drawable文件夹下

5.2 界面控件

• 5.2.2 Button和ImageButton

- 更改Button和ImageButton内容
- 引入android.widget.Button和android.widget.ImageButton

```
1. Button button = (Button)findViewById(R.id.Button01);  
2. ImageButton imageButton =  
   (ImageButton)findViewById(R.id.ImageButton01);  
3. button.setText("Button按钮");  
4. imageButton.setImageResource(R.drawable.download);
```

- 第1行代码用于引用在XML文件中定义的Button控件
- 第2行代码用于引用在XML文件中定义的ImageButton控件
- 第3行代码将Button的显示内容更改为 “Button按钮”
- 第4行代码利用setImageResource()函数，将新加入的png文件R.drawable.download传递给ImageButton



// 控件点击事件

Button控件设置点击事件的方式有三种，具体如下所示。

(1) 在布局文件中指定onClick属性的方式设置点击事件

```
<Button
.....
    android:onClick="click" />
```

(2) 使用匿名内部类的方式设置点击事件

```
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //实现点击事件的代码
    }
});
```



控件点击事件



(3) Activity实现OnClickListener接口的方式设置点击事件

```
public class Activity extends AppCompatActivity implements View.OnClickListener{  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        .....  
        btn.setOnClickListener(this); //设置Button控件的点击监听事件  
    }  
    @Override  
    public void onClick(View view) {  
        //实现点击事件的代码  
    }  
}
```

我们通过一个案例来讲解如何以三种方式为按钮设置点击事件。本案例的界面效果如下图所示。

- 1 **创建程序：**
 - ① 创建一个名为Button的应用程序
 - ② 指定包名为com.<yourname>.button

- 2 **放置界面控件：** 放置3个Button控件

- 3 **实现按钮的点击事件：** 在MainActivity中分别采用三种方式实现点击事件

- 4 **运行结果：** 运行程序，依次点击界面上的3个按钮，发现按钮上的文本信息都发生了变化



5.2 界面控件

• 5.2.2 Button和ImageButton

- 按钮响应点击事件：添加点击事件的监听器

```
1. final TextView textView = (TextView)findViewById(R.id.TextView01);  
2. button.setOnClickListener(new View.OnClickListener() {  
3.     public void onClick(View view) {  
4.         textView.setText("Button按钮");  
5.     }  
6. });  
7. imageButton.setOnClickListener(new View.OnClickListener() {  
8.     public void onClick(View view) {  
9.         textView.setText("ImageButton按钮");  
10.    }  
11. });
```

- 第2行代码中button对象通过调用setOnClickListener()函数注册一个点击(Click)事件的监听器View.OnClickListener()
- 第3行代码是点击事件的回调函数
- 第4行代码将TextView的显示内容更改为 “Button按钮”

5.2 界面控件

• 5.2.2 Button和ImageButton

- View.OnClickListener()
 - View.OnClickListener()是View定义的点击事件的监听器接口，并在接口中仅定义了onClick()函数
 - 当Button从Android界面框架中接收到事件后，首先检查这个事件是否是点击事件，如果是点击事件，同时Button又注册了监听器，则会调用该监听器中的onClick()函数
 - 每个View仅可以注册一个点击事件的监听器，如果使用setOnClickListener()函数注册第二个点击事件的监听器，之前注册的监听器将被自动注销

5.2 界面控件

• 5.2.2 Button和ImageButton

- 多个按钮注册到同一个点击事件的监听器上，代码如下：

```
1. Button.OnClickListener buttonListener = new Button.OnClickListener(){  
2.     @Override  
3.     public void onClick(View v) {  
4.         switch(v.getId()){  
5.             case R.id.Button01:  
6.                 textView.setText("Button按钮");  
7.                 return;  
8.             case R.id.ImageButton01:  
9.                 textView.setText("ImageButton按钮");  
10.                return;  
11.            }  
12.        }  
13.    };  
14.    button.setOnClickListener(buttonListener);  
15.    imageButton.setOnClickListener(buttonListener);
```

- 第1行至第12行代码定义了一个名为buttonListener的点击事件监听器
- 第13行代码将该监听器注册到Button上
- 第14行代码将该监听器注册到ImageButton上

// ImageView控件



ImageView控件表示图片，它继承自View，可以加载各种图片资源。ImageView控件的常用属性如下表所示。

属性名称	功能描述
android:layout_width	设置ImageView控件的宽度
android:layout_height	设置ImageView控件的高度
android:id	设置ImageView控件的唯一标识
android:background	设置ImageView控件的背景
android:layout_margin	设置当前控件与屏幕边界或周围控件的距离
android:src	设置ImageView控件需要显示的图片资源
android:scaleType	将图片资源缩放或移动，以适应ImageView控件的宽高
android:tint	将图片渲染成指定的颜色

// 练习



我们通过一个案例来讲解如何使用ImageView控件显示图片，本案例的界面效果如下图所示。

1

创建程序：

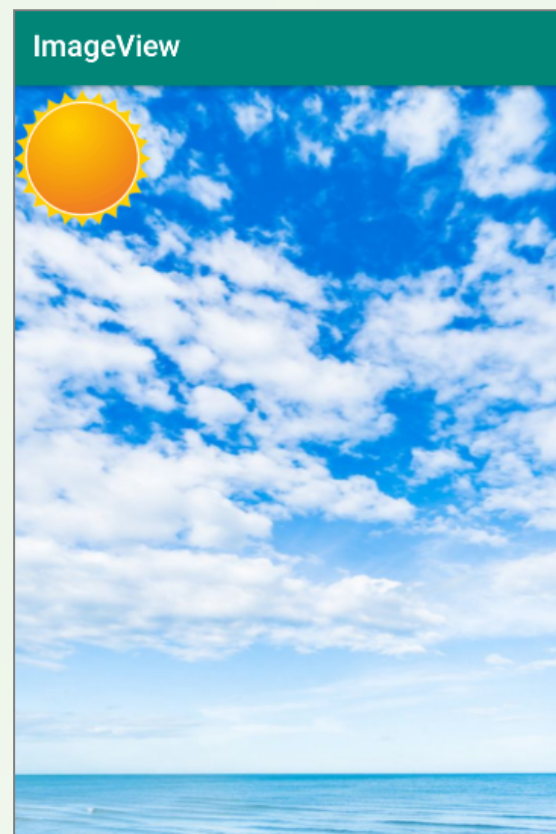
① 创建一个名为ImageView的应用程序
② 指定包名为com.<yourname>.imageview

2

导入界面图片：将界面需要的图片导入到drawable-hdpi文件夹中

3

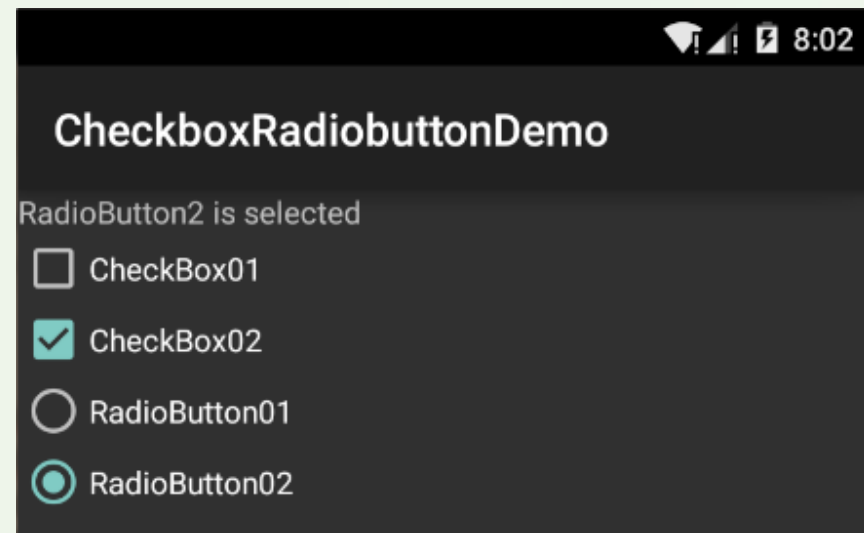
放置界面控件： 放置2个ImageView控件



5.2 界面控件

• 5.2.3 CheckBox和RadioButton

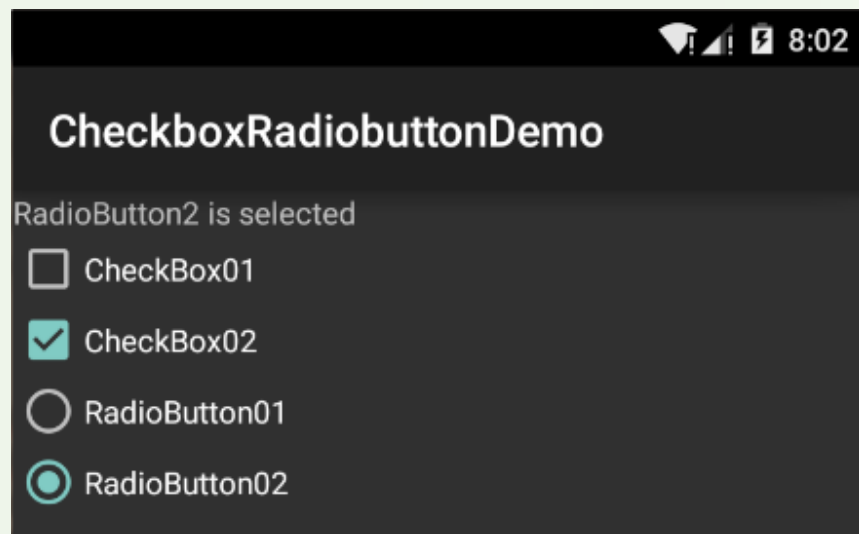
- CheckBox同时可以选择多个选项的控件
- RadioButton则是仅可以选择一个选项的控件
- RadioGroup是RadioButton的承载体，程序运行时不可见。应用程序中可能包含一个或多个RadioGroup，一个RadioGroup包含多个RadioButton，在每个RadioGroup中，用户仅能够选择其中一个RadioButton



5.2 界面控件

• 5.2.3 CheckBox和RadioButton

- 建立一个“CheckboxRadiobuttonDemo”工程，包含五个控件，从上至下分别是
 - TextView01
 - CheckBox01
 - CheckBox02
 - RadioButton01
 - RadioButton02



5.2 界面控件

• 5.2.3 CheckBox和RadioButton

- CheckboxRadiobuttonDemo在XML文件中的代码

```
1. <TextView android:id="@+id/TextView01"  
2.     android:layout_width="match_parent"  
3.     android:layout_height="wrap_content"  
4.     android:text="@string/hello"/>  
5. <CheckBox android:id="@+id/CheckBox01"  
6.     android:layout_width="wrap_content"  
7.     android:layout_height="wrap_content"  
8.     android:text="CheckBox01" >  
9. </CheckBox>  
10. <CheckBox android:id="@+id/CheckBox02"  
11.     android:layout_width="wrap_content"  
12.     android:layout_height="wrap_content"  
13.     android:text="CheckBox02" >  
14. </CheckBox>  
15. <RadioGroup android:id="@+id/RadioGroup01"  
16.     android:layout_width="wrap_content"  
17.     android:layout_height="wrap_content">  
18.     <RadioButton android:id="@+id/RadioButton01"  
19.         android:layout_width="wrap_content"  
20.         android:layout_height="wrap_content"  
21.         android:text="RadioButton01" >  
22.     </RadioButton>  
23.     <RadioButton android:id="@+id/RadioButton02"  
24.         android:layout_width="wrap_content"  
25.         android:layout_height="wrap_content"  
26.         android:text="RadioButton02" >  
27.     </RadioButton>  
28. </RadioGroup>
```

5.2 界面控件

• 5.2.3 CheckBox和RadioButton

- 引用CheckBox和RadioButton的方法参考下面的代码

```
1.  CheckBox checkBox1= (CheckBox)findViewById(R.id.CheckBox01);  
2.  RadioButton radioButton1 =(RadioButton)findViewById(R.id.RadioButton01);
```

- CheckBox设置点击事件监听器的简要代码

```
1.  CheckBox.OnClickListener checkboxListener = new CheckBox.OnClickListener(){  
2.  @Override  
3.  public void onClick(View v) {  
4.      //过程代码  
5.  }  
6.  };  
7.  checkBox1.setOnClickListener(checkboxListener);  
    checkBox2.setOnClickListener(checkboxListener);
```

- 与Button设置点击事件监听器中介绍的方法相似，唯一不同在于将Button.OnClickListener换成了CheckBox.OnClickListener

5.2 界面控件

• 5.2.3 CheckBox和RadioButton

- RadioButton设置点击事件监听器的方法

```
1.  RadioButton.OnClickListener radioButtonListener = new RadioButton.OnClickListener(){  
2.      @Override  
3.      public void onClick(View v) {  
4.          //过程代码  
5.      }  
6.      radioButton1.setOnClickListener(radioButtonListener);  
7.      radioButton2.setOnClickListener(radioButtonListener);  
}
```

// Toast类



Toast是Android系统提供的轻量级信息提醒机制，用于向用户提示即时消息，它显示在应用程序界面的最上层，显示一段时间后自动消失不会打断当前操作，也不获得焦点。

使用Toast显示提示信息的示例代码如下：

```
Toast.makeText(Context,Text,Time).show();
```

关于makeText()方法中参数的相关介绍具体如下：

- Context：表示应用程序环境的信息，即当前组件的上下文环境。
- Text：表示提示的字符串信息。
- Time：表示显示信息的时长，其属性值包括Toast.LENGTH_SHORT和Toast.LENGTH_LONG，分别表示显示较短时间和较长时间。

实战演练—实现注册界面效果

接下来我们通过实现一个注册界面的功能来演示如何使用Android程序中常用的简单控件，注册界面的效果如下图所示。

- 1 创建程序：**
 - ① 创建一个名为Register的应用程序
 - ② 指定包名为cn.itcast.register
- 2 导入界面图片：**将注册界面需要的图片导入到drawable-hdpi文件夹中
 - ① 创建分割线样式
 - ② 创建文本样式
- 3 创建样式：**
 - ③ 创建输入框样式
- 4 放置界面控件：**
 - ① 放置9个TextView控件、8个View控件
 - ② 放置1个ImageView控件、3个EditText控件
 - ③ 放置2个RadioButton控件、3个CheckBox控件
 - ④ 放置1个Button控件



The image shows a registration interface with a blue header labeled '注册'. Below the header are two buttons: '用QQ注册' (Use QQ to register) and '用微信注册' (Use WeChat to register). Below these is a button with an envelope icon labeled '使用电子邮箱注册' (Use email to register). The form fields are: '名字' (Name) with the value '小花', '邮箱' (Email) with the value 'xiaohua@163.com', and '密码' (Password) with masked characters '.....'. Below the password field are radio buttons for '性别' (Gender): '男' (Male) and '女' (Female), with '女' selected. Below the gender field are checkboxes for '请选择兴趣爱好' (Please select your hobbies): '唱歌' (Singing), '跳舞' (Dancing), and '读书' (Reading), with '唱歌' and '跳舞' selected. At the bottom is a '提交' (Submit) button.

实战演练—实现注册界面效果

5 去掉默认标题栏：修改theme属性的值去掉默认标题栏

① 获取界面控件创建文本样式

6 实现注册功能：② 设置单选按钮的点击事件

① 运行上述程序，并在界面上输入注册信息

7 运行结果：② 点击“提交”按钮

A screenshot of a registration form titled "注册" (Registration) in a blue header bar. The form has a light blue background with rounded corners. It features two columns of buttons at the top: "用QQ注册" (Register with QQ) and "用微信注册" (Register with WeChat). Below these is a row with an email icon and the text "使用电子邮箱注册" (Register with email). The form contains several input fields: "名字" (Name) with the value "小花" (Xiao Hua), "邮箱" (Email) with the value "xiaohua@163.com", and "密码" (Password) with masked dots. There is a "性别" (Gender) section with radio buttons for "男" (Male) and "女" (Female), where "女" is selected. Below that is a "请选择兴趣爱好" (Please select your hobbies) section with checkboxes for "唱歌" (Singing), "跳舞" (Dancing), and "读书" (Reading), all of which are checked. At the bottom, there is a "注册成功" (Registration successful) message and a "提交" (Submit) button.